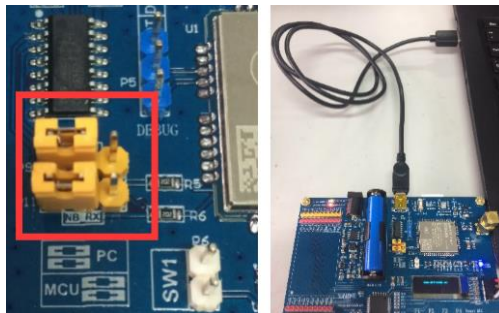


EVB_M1 串口调试 NB-IoT 入门篇

一、板子链接

如下图所示跳线帽接左侧两排排针表示 BC95 主串口连接 CH340 芯片，这时仅可以通过 mini USB 数据线连接 EVB_M1 和电脑，使用串口助手发送 AT 指令调试 BC95 模组，方便开发者熟悉 BC95 通信流程。



接下来，就使用我们的 EVB_M1 开发板，来讲解一下，如何入门 NB-IoT。

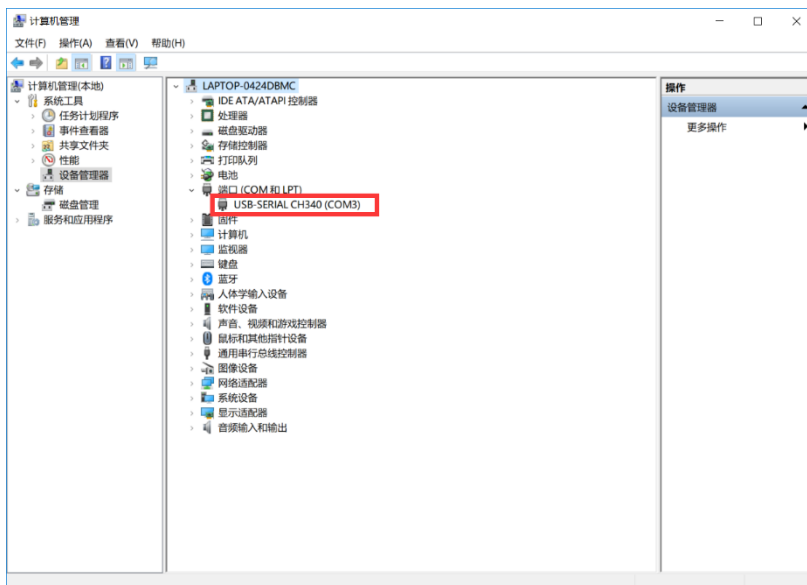
软件：

串口助手：QCOM_V1.6

二、驱动安装

首先将开发板上用数据线接入电脑的 USB 中，如果你电脑之前没有安装过 CH340 的驱动，会提醒你安装驱动，我们提供的工具包中，有“串口驱动-CH340_Windows”这个文件，解压缩，双击安装程序运行，一路下一步，直到安装完成。

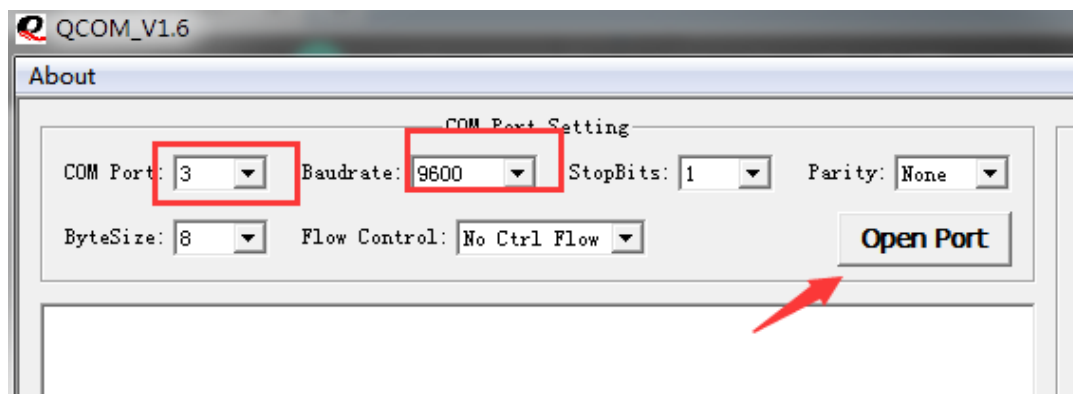
安装完之后，右击计算机（我的电脑），右键，管理，设备管理这栏，可以看到端口，这个时候有 CH340 字样的，后面紧跟的 COMx，就是串口号。示例中的是 COM3，这个接下来要用的到。



三、使用串口助手测试

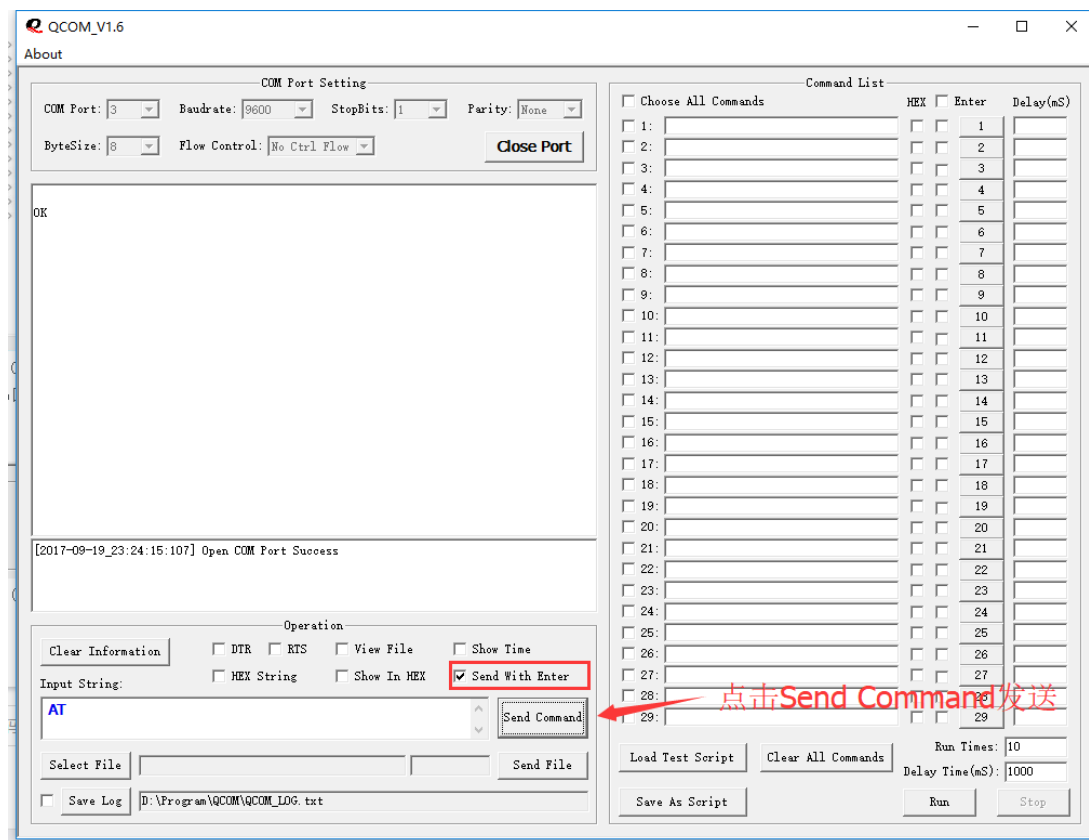
1、测试功能

这个时候可以打开串口助手，我们用到的工具是 Quetel 的 QCOM。



然后我们看到 COM Port 这个选项框中，有我们刚才看到的 1、3。这个是根据不同的电脑可能有不同的选项，具体要选择哪个数值，要和我们第一步中看到的端口一致。波特率（Baudrate）选择 9600 默认，其他的都默认选项，可以参照上图设置。然后点击 Open Port。（如果显示端口已占用，可以拔掉设备重新插一次）

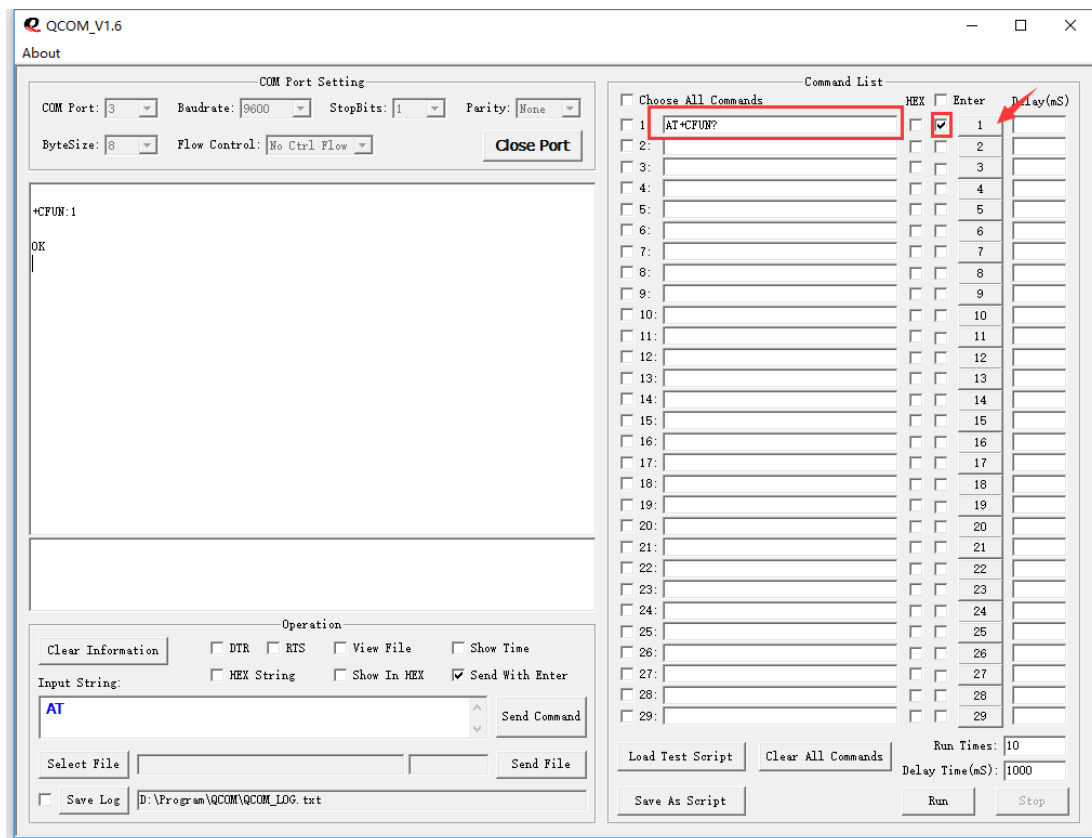
开始进行第一步，我们在 Input 这栏中，是我们要发送的内容，我们先验证一下：输入 [AT](#)



有很多朋友写完 [AT](#)，点击发送之后，模组没反应，这个时候就要看一下你是否勾选了 **Send With Enter**，因为模组将换行符作为 AT 指令的结束帧的，必须要勾选上，或者在输入框内敲入：Ctrl+Enter，点击发送（Send Command）这个时候在我们的返回框可以看到已经有返回 OK，说明我们的模组已正常工作了。

我们可以看到在右侧，有 1、2、3、4....28、29 这些框，这个是很便捷的一个功能，直接把我们的 AT 指令输入进去，点击发送后，软件会自动保存我们的输入记录，以便于下次继续使用。下面我们开始进行模组信息检测：

我们先输入“[AT+CFUN](#)”，然后选择 Enter 所对应的勾选框，点击对应的数字 1



这个时候看到模组有

[+CFUN:1](#)

[OK](#)

返回，如果你返回的是+CFUN:0，这个就是个错误的返回结果了，CFUN 的功能是射频开关，我们可以通过 [AT+CFUN=1](#) 来开启模组的射频开关，在 B656 版本之后，模组上电后默认是开启射频开关的，并不需要认为的设置 [AT+CFUN=1](#)，如果你查询 [AT+CFUN?](#) 返回的结果是 0，先等个几秒钟再查，如果依旧是 0 的话，就可能是以下几种情况：

- 1：模组频率设置不正确
- 2：模组 SIM 没有正常连接。

至于频率设的相关设置，大家可以点击：[BC95 频率设置](#) 来查看如何设置。如果模组的 SIM 卡没有识别到的话，那么 [AT+CFUN?](#) 的结果必然是 0，这个时候你要检查你的硬件电路是否有问题。

当 [AT+CFUN?](#) 返回结果为 1 的时候，这个时候我们才能进行下一步操作，

No1：首先我们检查一下卡的 IMSI

对应 AT 指令是：[AT+CIMI](#)，返回结果为你卡片所对应的 IMSI。

2、查询当前信号质量 CSQ

如果没有信号的话，自然是没办法连入网络，所以这一步我们检查一下我们的开发板是否连入网络。

输入：[AT+CSQ](#)，我这里返回结果为+CSQ:16,99。前面这个 16 就是信号质量，这个取值范围是 0-31 的一个数值，当这个数位 99 的时候，就说明没有获取到信号，这个时候我们可能需要多等一会儿，根据不同的地点，所等待时间在 1-60s 内，如果超过这个时间依旧返回的结果是+CSQ:99,99 的话，这个时候就要看一下卡是否是 NB 卡了，还有就是你的模组固件版本是否在 657 以下（之前版本没有开启扰码功能，不能成功接入基站）。还有一点，虽然你有很强的信号，但有时候入网入不了，这个时候你就要注意了，很有可能是你所处的环境信噪比太小，建议你换个地方尝试一下。

3、查询当前模组网络注册连接状态

虽然前面一步，我们已经获取到信号 CSQ 了，但这个时候不代表着已经可以测试这个网络了。因为 CSQ 仅仅代表着和基站交互没有问题，而我们想要上网，还是需要经过运营商核心网这条路的。入核心网有个很关键的地方，就是你使用的 NB SIM 卡，必须是 NB 专用卡，否则你的 IMSI 没有在核心网登记过，不会允许你进行下一步的。在卡正常的情况下，我们可以通过发送：

[AT+CEREG?](#)

这个指令返回的结果，可能有以下几种，这里详细的说一下：

[+CEREG:0,0](#)

[+CEREG:0,1](#)

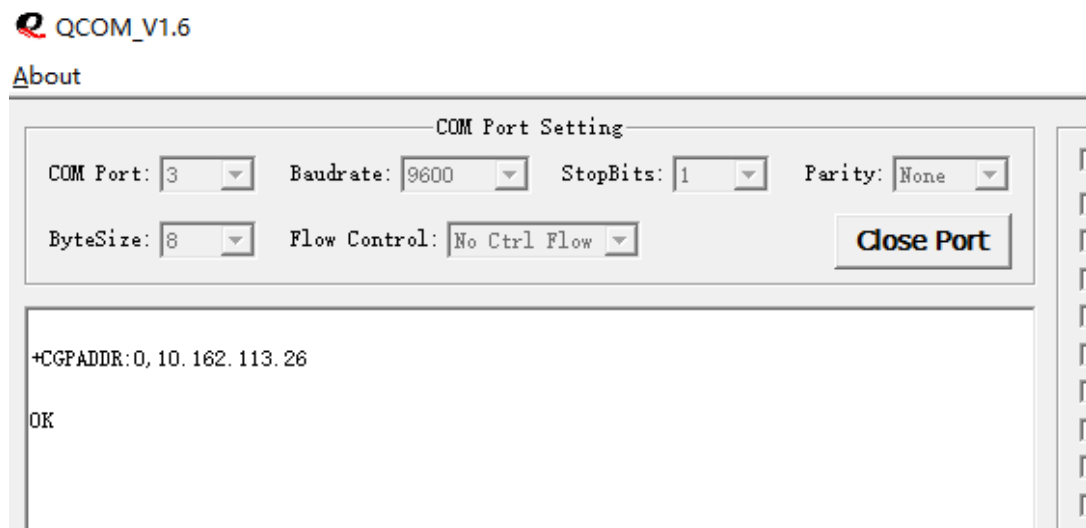
[+CEREG:0,2](#)

前面一个 0，是功能码，如果设置为 0，只有我们请求的时候才会返回+CEREG 这个结果，设为 1，一旦网络状态发生改变的时候，会自动上报 URC 来通知我们。

后面的 0,1,2，当为 0 的时候，说明网络还未注册，依旧在搜索信号，一般刚开机的时候，发送请求会返回为 0，当为 1 的时候，这个时候表明网络已经注册成功了，可以正常使用了。如果为 2 的时候，这个是从 0 到 2 的转换，再次尝试入网，这个时候就说明网络质量或者线路并不是很流畅，模组在尝试入网。如果一直为 2 的话，建议重启模组或重启射频 CFUN。直至返回结果为+CEREG:0,1。当然后面还有 3,4,5 等，这些目前都用不到。

4、查询当前模组当前的 IP 地址

对应 AT 指令：[AT+CGPADDR](#)。这步是多余的，但为什么讲这一步呢，因为很多人把这个 IP 地址当作寻呼模组的一个 IP 地址。我们先看一下返回结果



CGPADDR:0,10.162.113.26，从结果上看，10.162.113.26 是我们模组的 IP 地址，这么说没错，但，这个地址并不是公网地址，只是运营商网络给分配的一个内网地址，10 段 IP 一直都是用在内网上的，所以不要想了，即使你有回天之术，也不能在外网通过 10 段 IP 访问到模组。所以，虽然 COAP 支持 M2M，但 NB 模组在 IPV6 普及之前也无法使用的。

5、PING 服务器

对应 AT 指令：

[AT+NPING=X.X.X.X](#)

我个人觉得这步很关键，为什么呢？因为现在电信的 NB 是定向卡，如果你的服务器 IP 没有和 NB 卡进行绑定的话，是没法访问到服务器的，测试卡和服务器是否绑定，只需要发送一个 PING 命令，如果返回：+NPING:x.x.x.x,115,1042 字样字段，说明你可以拿这个服务器测试，如果不行，多试几次，还是不行的话，那就是你的 NB 卡并没有绑定这个服务器。注意如果你发送完直接就返回+NPINGERR:1，那就说明你的 NB 卡和服务器真没绑定。

6、创建 UDP Socket NSOCR

对应 AT 指令：[AT+NSOCR=DGRAM,17,8888,1](#)。前面两个参数可以忽略，按照这个来写就行，第三个 8888 是指模组所需要监听的 UDP PORT 端口，取值范围 1-65535 当然你也不需要记住，因为一旦通过运营商网络运营商会 NAT 转发，公网中就不是这个端口了。不过如果你要创建多个 socket 的话，注意这个参数不能重复。

返回：

0

OK

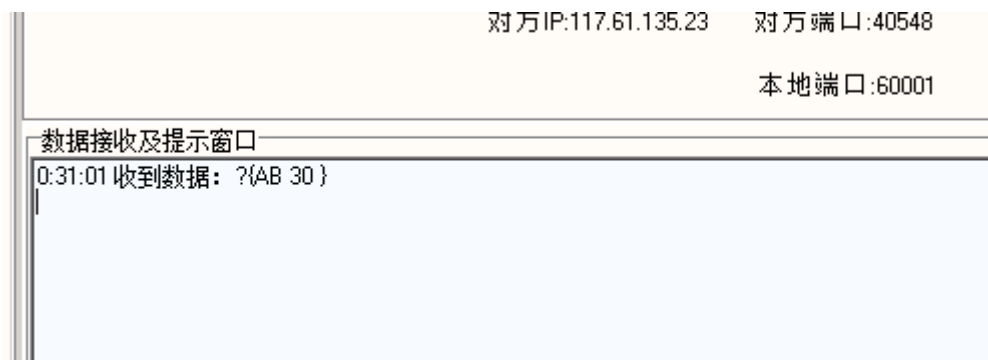
前面一个 0 就是 socket id 号，socket id 最多只能创建 0~6，也就是 7 个，如果超过了，就会返回 ERROR。这个 id 号也是我们后面发送数据及接收数据要用的，前期的话我们可以只创建一个试试。

7、发送 UDP 数据 NSOST

对应指令：[AT+NSOST=0,x.x.x.x,60001,2,AB30](#)。前面做那么多铺垫终于到了重头戏了，第一个参数 0，就是我们上面创建的 socket id 返回的 id 号，我们可以把它称为 0 号场景。

x.x.x.x 是我们的服务器 IP 地址，60001 是端口，根据自己服务器开的端口而定，2 是要发送的数据长度，AB30 是要发送的数据，因为数据是 16 进制的，所以就是 0xAB 0x30 这两个数据。

返回的结果：0,2 OK，其中 0 是 socket id，2 是发送的数据长度，这个时候我们的服务器应该就可以收到我们发送的数据了。



和我们发送的数据一模一样的。

8、接收 UDP 数据 （第一步 NSONMI）

这步没有对应的 AT 指令，当我们通过 No7 向服务器发送数据后，这个时候运营商为模组和服务器开通了一个短暂通道，模组的地址就是我们接收数据时候的地址，我的这个是：117.61.135.23，端口：40548，这个地址是临时的，正常有效时间不会超过 1 分钟，所以你最好在 30 秒内通过这个地址将数据发送给模组，一旦超过这个时间，链路就会失效，你必须再次用模组向服务器发送一个数据，请求新的链路。

我们在 UDP 调试工具中十六进制输入：010203，点击发送：

此时模组 URC 上报一个消息：+NSONMI:0,3，这个是告诉我们，socket id 为 0 的场景中收到一个字节长度为 3 的数据，此时我们并不能查看数据，必须通过下个步骤 No9。

9、接收 UDP 数据 （第二步 NSORF）

对应 AT 指令：AT+NSORF=0,3。实际上当我们在 No8 的步骤中的时候，数据已经到了我们的模组中，但华为为了让大家知晓有多少数据过来，方便开启 Ram 缓冲，所以分为两步走，这一步的作用就是把数据从模组中取出来。第一个参数 0 代表着 socket id，用来识别内容来自哪个 id 的，第二给 3 是长度，当然，你也可以大于 3，只要不小于 No8 得到的长度。在下个数据或重启之前，都是可以取出这个数据的，和时间没有关系。发送 AT+NSORF=0,3，返回：0,x.x.x.x,60001,3,010203,0，这就好理解了，x.x.x.x 是服务器地址，60001 是我开启的端口，3 是收到的数据长度，010203 是我刚才从服务器发送的十六进制数据。最后一个 0 是还没有读取的数据长度，因为我这里读取的是 3，把所有的数据读完了，所以显示为 0。

好了，9 个步骤，很安全的把 NB-IoT 的 UDP 演示一遍，如果做产品的话，其中一些可以省略。